# UNITED STATES PATENT AND TRADEMARK OFFICE

| APPLICATION NO. | FILING DATE | FIRST NAMED INVENTOR | ATTORNEY DOCKET NO. | CONFIRMATION NO. |
|---|---|---|---|---|
| 10/020,033 | 12/06/2001 | Attila D. Banki | PM 2000.063 | 8954 |

| | | |
|---|---|---|
| 7590 | 10/14/2005 | EXAMINER |

Gary D. Lawson
ExxonMobil Upstream Research Company
P.O. Box 2189
Houston, TX 77252-2189

PROCTOR, JASON SCOTT

| ART UNIT | PAPER NUMBER |
|---|---|
| 2123 | |

DATE MAILED: 10/14/2005

Please find below and/or attached an Office communication concerning this application or proceeding.

| | Application No. | Applicant(s) |
|---|---|---|
| **Office Action Summary** | 10/020,033 | BANKI ET AL. |
| | Examiner | Art Unit | |
| | Jason Proctor | 2123 | |

-- *The MAILING DATE of this communication appears on the cover sheet with the correspondence address* --

**Period for Reply**

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE <u>3</u> MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.
- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133).
  Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

**Status**

1)☒ Responsive to communication(s) filed on <u>28 July 2005</u>.

2a)☒ This action is **FINAL.**    2b)☐ This action is non-final.

3)☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle,* 1935 C.D. 11, 453 O.G. 213.

**Disposition of Claims**

4)☒ Claim(s) <u>1-34</u> is/are pending in the application.

    4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)☐ Claim(s) _____ is/are allowed.

6)☒ Claim(s) <u>1-34</u> is/are rejected.

7)☐ Claim(s) _____ is/are objected to.

8)☐ Claim(s) _____ are subject to restriction and/or election requirement.

**Application Papers**

9)☐ The specification is objected to by the Examiner.

10)☒ The drawing(s) filed on <u>06 December 2001</u> is/are: a)☒ accepted or b)☐ objected to by the Examiner.

    Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

    Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11)☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

**Priority under 35 U.S.C. § 119**

12)☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

    a)☐ All  b)☐ Some * c)☐ None of:

        1.☐ Certified copies of the priority documents have been received.

        2.☐ Certified copies of the priority documents have been received in Application No. _____.

        3.☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

    * See the attached detailed Office action for a list of the certified copies not received.

**Attachment(s)**

1)☒ Notice of References Cited (PTO-892)

2)☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)

3)☐ Information Disclosure Statement(s) (PTO-1449 or PTO/SB/08)
    Paper No(s)/Mail Date _____.

4)☐ Interview Summary (PTO-413)
    Paper No(s)/Mail Date. _____ .

5)☐ Notice of Informal Patent Application (PTO-152)

6)☐ Other: _____.

## DETAILED ACTION

Claims 1-34 are pending in the application. Claims 1-34 have been rejected.

The Examiner thanks Applicants for pointing out the unintentional oversight in the previous action regarding the rejections of claims 30 and 31. However, these claims were anticipated by the Bentley reference as relied upon to reject independent claim 20 under 35 U.S.C. § 102. In particular, Bentley anticipates the limitations of claim 30 (column 4, lines 32-42) while claim 31 reiterates the limitations of claim 19, which was specifically rejected. The Examiner apologizes for this clearly unintentional omission.

Applicants' request that the current action be non-final in response to the unintentional oversight in the previous action is counter to the interest of compact prosecution. The prior art relied upon in the previous rejection anticipates the claims at issue, Applicants' amendments necessitate new grounds of rejection regardless of how the claims would have been originally rejected, and Applicants have been afforded an interview with the Examiner on 12 July 2005, during which these claims were not discussed.

The Examiner apologizes for the omission, however reiterates that any arguments regarding claims 30 and 31 would have been moot in light of the amendments to the independent claims.

In the event that Applicants identify an omission in the current Office Action, the Examiner respectfully requests that Applicants notify the Examiner by telephone in order to promote compact prosecution.

### *Objections to the Specification*

The Examiner thanks Applicants for addressing the previous objections to the specification,

which have been withdrawn.

### *Rejections under 35 U.S.C. § 112*

The Examiner thanks Applicants for addressing the previous rejections under 35 U.S.C. § 112,

second paragraph, which have been withdrawn.

The following is a quotation of the second paragraph of 35 U.S.C. 112:

> The specification shall conclude with one or more claims particularly pointing out and distinctly claiming the
> subject matter which the applicant regards as his invention.

1.      Claims 32-34 are rejected under 35 U.S.C. § 112, second paragraph, as being indefinite

for failing to particularly point out and distinctly claim the subject matter which applicant

regards as the invention.

        Claim 34 recites a step of "integrate the custom object oriented code with the main

simulation system in a seamless manner that incorporates components and procedural algorithms

defined by the main simulation system with the custom components and procedural algorithms

defined by the simulation user into an integrated simulation, wherein the integrated simulation

represents a model of a unique physical system configuration" which fails to particularly point

out and distinctly claim the invention.  This limitation restricts the claimed invention to modeling

only unique physical system configurations.  Modeling a physical system that is not unique, for

example, a mass produced machine, is not covered by this claim language.  Because it this

limitation is not clearly disclosed in the specification, the metes and bounds of Applicants' claim

are indefinite. Further, it is unclear how to determine whether a prior art modeling system was

designed to model unique physical system configurations. This is a vague and indefinite

limitation of intended use which cannot reasonably be assessed against the prior art.

Claim 35 recites the phrase "graphical logic", the meaning of which is unclear. The

Examiner presumes Applicants intend "graphic logic flow chart" or some equivalent.

Claim 36 stands rejected by virtue of its dependence.

### *Claim Rejections - 35 USC § 103*

The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness

rejections set forth in this Office action:

> (a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in
> section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are
> such that the subject matter as a whole would have been obvious at the time the invention was made to a person
> having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the
> manner in which the invention was made.

The factual inquiries set forth in *Graham* v. *John Deere Co.*, 383 U.S. 1, 148 USPQ 459 (1966),

that are applied for establishing a background for determining obviousness under 35

U.S.C. 103(a) are summarized as follows:

1. Determining the scope and contents of the prior art.
2. Ascertaining the differences between the prior art and the claims at issue.
3. Resolving the level of ordinary skill in the pertinent art.
4. Considering objective evidence present in the application indicating obviousness or nonobviousness.

2. Claims 1-34 are rejected under 35 U.S.C. § 103(a) as being unpatentable over US Patent

No. 4,901,221 to Kodosky et al. (Kodosky '221) in view of "Object-Oriented Analysis and

Simulation" by David R. C. Hill (Hill).

Regarding claims 1, 20, and 32, Kodosky '221 teaches a computer system for simulating a physical system [(abstract); *"The present invention comprises a novel system and associated method for modeling a process."* (column 7, lines 21-35); *"The generalized diagram of FIG. 4 shows an instrumentation system 56 incorporating the second system 28 shown in FIG. 2"; "The instrumentation system 56 can be used to control the instrument 60, to acquire data from the instrument 60 to analyze that data, to store that data and to present that data."* (column 8, lines 24-64); FIGS. 2 and 4; The system models a process that can be controlled via instrumentation, thus a physical process.]

Kodosky '221 teaches an interface that enables a simulator user to dynamically construct logic to customize simulation of the physical system [*"implementation of the data structure like that of the diagram of FIG. 19a advantageously permits the implementation of a system in which execution instructions can be constructed in a graphical fashion. [...] Thus, a user need only construct an appropriate visual display in order to construct the execution instructions."* (column 13, lines 43-60)], incorporating custom components and procedural algorithms that extend functionality of a main simulation system [demonstrated in FIGS. 20A-20L, corresponding description at (column 14, line 62 – column 16, line 2)].

Kodosky '221 teaches initiating and executing the simulation [*"The system also includes an execution subsystem for assigning respective values for the one or more input variables and for executing the execution instructions to produce respective values for the one or more output variables."* (column 3, lines 50-65); also (column 16, line 3 – column 17, line 2)]. Kodosky discloses no required intervention by the user.

Kodosky '221 teaches integrating the simulation objects (*virtual instruments*) to produce

an integrated customized simulation system without intervention of the user [ *"In response to the*

*step of electronically constructing the diagram display* [performed by the user], *execution*

*instructions are electronically constructed which characterize an execution procedure which*

*substantially corresponds to the displayed procedure.* [...] *The execution instructions are*

*electronically executed to produce respective values for respective output variables.* " (column 4,

lines 2-14)].

Although Kodosky '221 does not explicitly state that the execution instructions are

"object-oriented", the execution instructions are clearly functionally equivalent, especially in the

context of the claimed invention. Further, a person of ordinary skill in the art would recognize

that the form of a program's source code, whether object-oriented, structured programming,

functional programming, or other, are all functionally equivalent when executed by the machine.

The limitations not explicitly taught by Kodosky '221, such as "means for converting the

constructing logic code into corresponding object-oriented code <u>during a simulation without</u>

<u>intervention of the simulation user</u>" and other limitations relating to manipulating the simulation

while it is executing is taught by the concept well-known in the art by the name "visual

interactive simulation". Hill teaches this concept on page 138:

> The third technique, called interactive visual simulation, allows modifications of the model during
> execution, in order to display immediately the effects caused by these changes.

Hill elaborates on pages 142-144, including:

> Despite these drawbacks, it should be pointed out that the involvement of users who are experts in the
> system modeled is, as we have already mentioned, far greater if they can use the 'commands' of the model
> themselves. [...] Because of their knowledge of the system, experts contribute tremendous skill at the
> model validation level. What is more, when it comes to the analysis and experimentation of new strategies,
> the experts themselves can make proposals directly and so take over the new modified model which in a
> way becomes their model. Visual interactive simulation therefore brings about a greater involvement of
> experts in the simulation project.

Whether Applicants' claimed invention involves animating the simulation (visual simulation) or not, a person of ordinary skill in the art at the time of Applicants' invention would have found it obvious to implement an interactive feature in the system of Kodosky '221, directed by Hill's ample motivation. To be more specific, it would have been obvious to a person of ordinary skill in the art at the time of Applicants' invention, in response to the explicit teachings of Hill regarding interactive simulation and the benefits of interactive simulation, to combine the concept of interactive simulation with the modeling system of Kodosky '221 to arrive at a modeling system that performs an interactive simulation. As taught by Hill, such a system would allow a simulation user, such as "an expert in the system [being] modeled" rather than a computer programmer, to 'command' the model while the simulation is being executed.

Specifically regarding claim 20, where a user is a person of ordinary skill in the art of reservoirs, it would have been obvious to use the combined teachings of Kodosky '221 and Hill to simulate a reservoir as recited by the claim.

For Applicants' convenience, the Examiner refers Applicants to US Patent No. 5,974,254 to Hsu, which is not relied upon to form the above rejection but affirms the Examiner's interpretation of the Kodosky '221 patent. From Hsu, column 2, line 34 – column 3, line 25:

> U.S. Pat. No. 4,901,221 to Kodosky et al discloses a graphical system and method for modeling a process, i.e. a graphical programming environment which enables a user to easily and intuitively model a process. The graphical programming environment disclosed in Kodosky et al can be considered the highest and most intuitive way in which to interact with a computer. A graphically based programming environment can be represented at level above text-based high level programming languages such as C, Pascal, etc. The method disclosed in Kodosky et al allows a user to construct a diagram using a block diagram editor such that the diagram created graphically displays a procedure or method for accomplishing a certain result, such as manipulating one or more input variables to produce one or more output variables. As the user constructs the data flow diagram using the block diagram editor, machine language instructions are automatically constructed which characterize an execution procedure which corresponds to the displayed procedure. Therefore, a user can create a text-based computer program solely by using a graphically based

programming environment. This graphically based programming environment may be used for creating virtual instrumentation systems and modeling processes as well as for any type of general programming.

Therefore, Kodosky et al teaches a graphical programming environment wherein a user manipulates icons in a block diagram using a block diagram editor to create a data flow "program." A graphical program for controlling instruments or implementing instrumentation functions is referred to as a virtual instrument (VI). In creating a virtual instrument, a user preferably first creates a front panel including various controls or indicators that represent the respective input and output that will be used by the VI. When the controls and indicators are created in the front panel, corresponding icons or terminals are automatically created in the block diagram by the block diagram editor. Alternatively, the user may first place data input/output icons in the block diagram, which may optionally cause control and indicator icons to be placed in a front panel. The user then chooses various functions that accomplish his desired result, connecting the corresponding function icons between the terminals of the respective controls and indicators. In other words, the user creates a data flow and/or control flow program, referred to as a block diagram, representing the graphical data flow which accomplishes his desired function. This is done by wiring up the various function icons between the control icons and indicator icons. The manipulation and organization of icons in turn produces machine language that accomplishes the desired method or process as shown in the block diagram.

A user inputs data to a virtual instrument using front panel controls. This input data propagates through the data flow block diagram or graphical program and appears as changes on the output indicators. The data that flows from the controls to the indicators in this manner is referred to as control data. In an instrumentation application, the front panel can be analogized to the front panel of an instrument. The user adjusts the controls on the front panel to affect the input and views the output on the respective indicators.

Regarding claim 2, Kodosky '221 teaches that the constructed logic comprises facility management logic which is representative of steps for monitoring and controlling mechanical facilities associated with the physical system [*"The instrumentation system 56 can be used to control the instrument 60, to acquire data from the instrument 60 to analyze that data, to store that data and to present that data."* (column 8, lines 38-64); A person of ordinary skill in the art would recognize an instrument which can be controlled or used to acquire data as a "mechanical facility".]

Regarding claim 23, Kodosky '221 teaches that construction of the logic comprises using a graphical user interface to perform the step of selecting and using an existing logic, selecting and modifying existing logic, or developing new logic [*"execution instructions can be*

*constructed by constructing a visual display in which at least one input variable produces at least output variable according to a displayed procedure. [...] Thus, a user need only construct an appropriate visual display in order to construct the execution instructions."* (column 13, lines 43-60)]. Hsu affirms this interpretation [*"Kodosky et al teaches a graphical programming environment wherein a user manipulates icons in a block diagram using a block diagram editor to create a data flow "program"."* (column 2, lines 58-61)].

Regarding claims 3 and 24, Kodosky '221 teaches that the logic interface comprises a logic flow chart interface [FIG 19A; *"FIG. 19a shows a system representation of a virtual instrument."* (column 13, lines 67 – column 14, line 23)]. This interpretation is supported by Hsu [*"As the user constructs the data flow diagram using the block diagram editor..."* (column 2, lines 48-57) A person of ordinary skill in the art would recognize a "data flow diagram" as a teaching of a "logic flow chart".]

Regarding claim 25, Kodosky '221 teaches that the construction of the logic produces a text-based logic code [*"execution instruction can be constructed by constructing a visual display"* (column 13, lines 43-60)]. Hsu affirms this interpretation [*"As the user constructs the data flow diagram using the block diagram editor, machine language instructions are automatically constructed which characterize an execution procedure which corresponds to the displayed procedure."* (column 2, lines 48-52)].
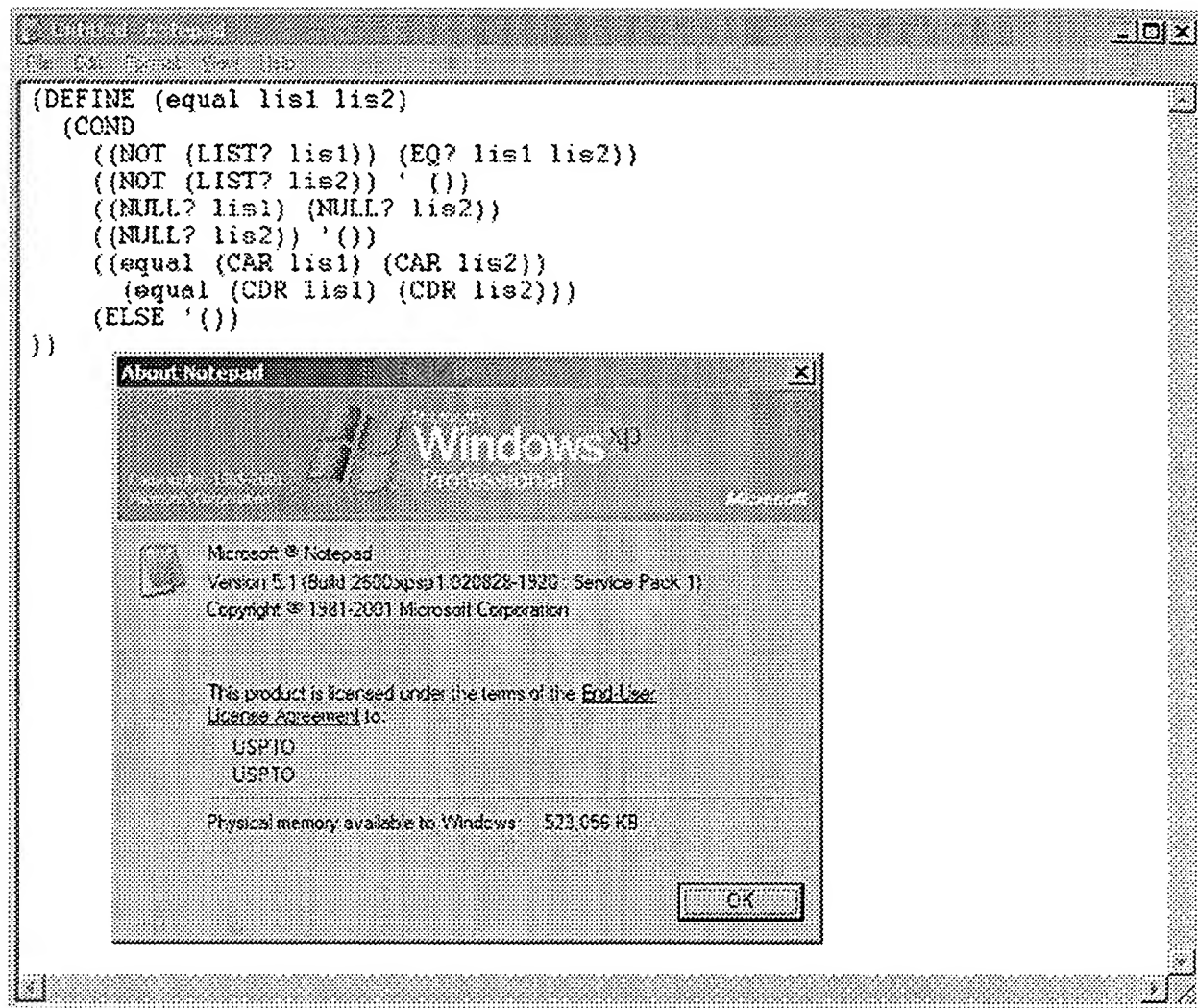
Regarding claims 4, 5, 26, and 33, Kodosky '221 teaches that the logic flow chart interface comprises one or more of icons, arrows, menus, dialogs, toolbar buttons, and text to enable the simulator user to build up, edit, and visualize the facility management logic in the form of a flow chart [*"To get started, double click on the PROTO icon.* [...] *You are now ready to build your virtual instrument."* (column 22, lines 46-57); FIG. 57 depicts a flow chart of the same system]. Kodosky '221 further discloses icons representing basic logic control constructs for looping, decision making, statement execution, and logic entry and exit [FIGS. 8-12 and corresponding description; FIG. 22 and description at column 17, lines 15-68).

Regarding claim 6, Kodosky '221 teaches that icons that represent logic control mechanisms enable the simulator user of the computer system to construct customized logic flow charts [FIGS. 8-12 and corresponding description; FIGS. 20E-20L corresponding description at (column 14, line 62 – column 16, line 2)].

Regarding claims 7, 8, and 27, Official notice is taken that a graphical interface including a text-based interface for providing logic code is extremely well known in the art. In Applicants' response, Applicants seasonably traverse and challenge the Examiner's use of Official Notice. The Examiner respectfully submits the following screen shot of Microsoft® Notepad, copyright 1981-2001, as "a graphical interface including a text based interface for providing logic code". The Examiner reasserts that it is "well known" to use "a graphical interface including a text based interface for providing logic code".

```
(DEFINE (equal lis1 lis2)
  (COND
    ((NOT (LIST? lis1)) (EQ? lis1 lis2))
    ((NOT (LIST? lis2)) ' ())
    ((NULL? lis1) (NULL? lis2))
    ((NULL? lis2)) '())
    ((equal (CAR lis1) (CAR lis2))
       (equal (CDR lis1) (CDR lis2)))
    (ELSE '())
))
```

Regarding claim 9, Kodosky '221 does not expressly disclose that the text-based logic

code is capable of being automatically created from a logic flow chart. However, this limitation

does not positively recite that the logic code is actually created from a logic flow chart. Kodosky

'221 does not explicitly disclose that the code (inherent and required to execute the simulation)

cannot be created from a logic flow chart and therefore teaches this limitation. The Examiner

respectfully suggests that Applicants' submit claim language that positively defines the claimed

invention.

Claim 10 similarly recites a limitation using the word "capable". Kodosky '221 does not explicitly disclose that the language of the code (inherent and required to execute the simulation) is not convertible into "object oriented facility management code in the form of C++".

Regarding similar limitations in claim 28, Kodosky '221 teaches that the model is converted into executable instructions (column 4, lines 5-14). Hsu affirms this interpretation ["*As the user constructs the data flow diagram using the block diagram editor, machine language instructions are automatically constructed which characterize an execution procedure which corresponds to the displayed procedure.*" (column 2, lines 34-58)]. A person of ordinary skill in the art would recognize the functional equivalence of the teachings in Kodosky '221 and "C++ code". The functional equivalence of various programming languages is well known to artisans of ordinary skill.

Regarding claims 11 and 12, Kodosky '221 teaches that the simulation is executed ["*In response to the step of electronically constructing the diagram display* [performed by the user], *execution instructions are electronically constructed which characterize an execution procedure which substantially corresponds to the displayed procedure.* [...] *The execution instructions are electronically executed to produce respective values for respective output variables.*" (column 4, lines 2-14)]. Although Kodosky '221 does not explicitly state that the execution instructions are "object-oriented", the execution instructions are clearly functionally equivalent, especially in the context of the claimed invention. Further, a person of ordinary skill in the art would recognize

that the form of a program's source code, whether object-oriented, structured programming, functional programming, or other, are all functionally equivalent when executed by the machine.

Regarding claim 13, Kodosky '221 teaches that the system enables the user to develop logic using a logic flow chart interface [*"implementation of the data structure like that of the diagram of FIG. 19a advantageously permits the implementation of a system in which execution instructions can be constructed in a graphical fashion. [...] Thus, a user need only construct an appropriate visual display in order to construct the execution instructions."* (column 13, lines 43-60)]. This interpretation is supported by Hsu [*"The method disclosed in Kodosky et al allows a user to construct a diagram using a block diagram editor such that the diagram created graphically displays a procedure or method for accomplishing a certain result"* (column 2, lines 43-48)].

Claims 14 and 29 recite limitations using the words "capability" and "enabling". Kodosky '221 does not explicitly disclose that the code (inherent and required to execute the simulation) is incapable of extending the simulation data model by creating new classes that inherit from the simulation data model, thereby enabling the object-oriented code to call functions of the simulation data model and use member data of the simulation data model. Although Kodosky '221 does not explicitly state that the execution instructions are "object-oriented", the execution instructions are clearly functionally equivalent, especially in the context of the claimed invention. Further, a person of ordinary skill in the art would recognize that the

form of a program's source code, whether object-oriented, structured programming, functional programming, or other, are all functionally equivalent when executed by the machine.

Claims 15 and 16 recite limitations typical of any object-oriented programming language. See, for example, "Java API Documentation", copyright 1996, provided for Applicants' convenience. Although Kodosky '221 does not explicitly state that the execution instructions are "object-oriented", the execution instructions are clearly functionally equivalent, especially in the context of the claimed invention. Further, a person of ordinary skill in the art would recognize that the form of a program's source code, whether object-oriented, structured programming, functional programming, or other, are all functionally equivalent when executed by the machine.

Regarding claims 17 and 18, Kodosky '221 does not explicitly teach invoking the code at a plurality of timesteps during the simulation or returns control back to the main simulation system after the facility management code has finished executing. However, these limitations are clearly inherent in the disclosure that "the invention provides a system for modeling a process" (column 3, lines 50-65) and subsequent disclosure regarding the operation of the system.

Regarding claims 19 and 31, Official Notice is taken that plural processor systems are known in the art, their advantages are known in the art, and a person of ordinary skill in the art would have found it obvious to implement the combined teachings of Kodosky '221 and Hill on a plural processor system. Numerous references for performing a simulation on a plural

processor system can be found in the prior art, including at least Japanese Patent Publication No.

JP 06035863 A, which has been provide with the English language abstract.

Claims 21 and 22 further recite the intended use of modeling a reservoir as recited by

claim 20. As the combined teachings of Kodosky '221 and Hill would have rendered it obvious

to a person of ordinary skill in the art of reservoirs to model and simulate a reservoir, it further

would have been obvious to that person of ordinary skill in the art to model a "hydrocarbon-

bearing subterranean formation" or equivalent terminology for a reservoir. It further would have

been obvious to that person of ordinary skill in the art to model and simulate a "fluid-containing

facilities associated with production of hydrocarbons from the hydrocarbon-bearing subterranean

formation" or equivalent terminology for the specific "physical system".

Regarding claim 30, Kodosky '221 teaches that the system models a process, including

producing output variables (abstract), therefore Kodosky '221 teaches generating results for

predicting the overall behavior of the physical system.

Regarding claim 34, Kodosky '221 teaches an interactive interface [*"A process typically

can be characterized by a reception of one or more input variables and a provision of one or

more output variables."* (column 3, line 66 – column 4, line 14)]. Hsu affirms this interpretation

[*"The graphical programming environment disclosed in Kodosky et al can be considered the

highest and most intuitive way in which to interact with a computer. A graphically based

programming environment..."* (column 2, lines 34-58)].

*Conclusion*

Applicant's amendment necessitated the new ground(s) of rejection presented in this

Office action. Accordingly, **THIS ACTION IS MADE FINAL**. See MPEP § 706.07(a).

Applicant is reminded of the extension of time policy as set forth in 37 CFR 1.136(a).

A shortened statutory period for reply to this final action is set to expire THREE

MONTHS from the mailing date of this action. In the event a first reply is filed within TWO

MONTHS of the mailing date of this final action and the advisory action is not mailed until after

the end of the THREE-MONTH shortened statutory period, then the shortened statutory period

will expire on the date the advisory action is mailed, and any extension fee pursuant to 37

CFR 1.136(a) will be calculated from the mailing date of the advisory action. In no event,

however, will the statutory period for reply expire later than SIX MONTHS from the date of this

final action.

Art considered pertinent by the examiner but not applied has been cited on form PTO-

892.

Any inquiry concerning this communication or earlier communications from the

examiner should be directed to Jason Proctor whose telephone number is (571) 272-3713. The

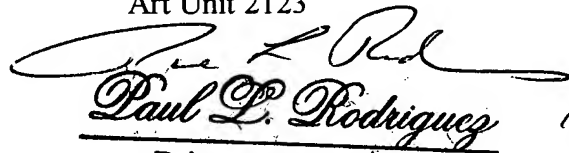examiner can normally be reached on 8:30 am-4:30 pm M-F.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Leo Picard can be reached at (571) 272-3749. The fax phone number for the organization where this application or proceeding is assigned is (571) 273-8300.

Any inquiry of a general nature or relating to the status of this application should be directed to the TC 2100 Group receptionist: 571-272-2100. Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see http://pair-direct.uspto.gov. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free).

Jason Proctor
Examiner
Art Unit 2123

Paul L. Rodriguez    10/12/05
Primary Examiner
Art Unit 2125

jsp